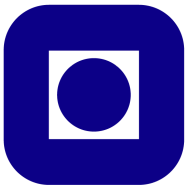

NTNU

TDT4136 - Introduction to Artificial Intelligence

Exercise 3

Mathias Ose & Øyvind Robertsen

October 3, 2014

1 Using the A* Algorithm

1.1 Problem A: Pathfinding in 2D Games

All deliverables in the form of code are included aside the report.

Subproblem A.1: Grids with Obstacles

The following figures show the solution path calculated by the A* implementation, with the OPEN and CLOSED sets visualized as respectively blue and red dots.

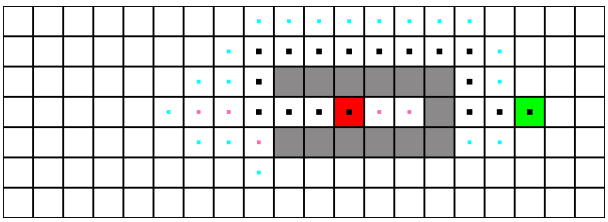

Figure 1.1: Board 1.1

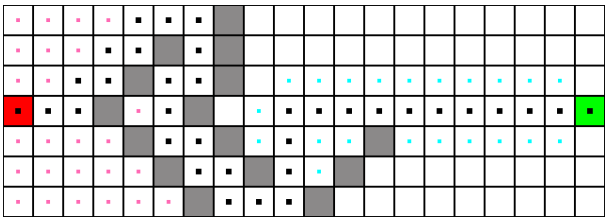

Figure 1.2: Board 1.2

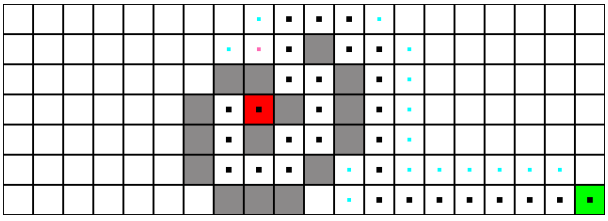

Figure 1.3: Board 1.3

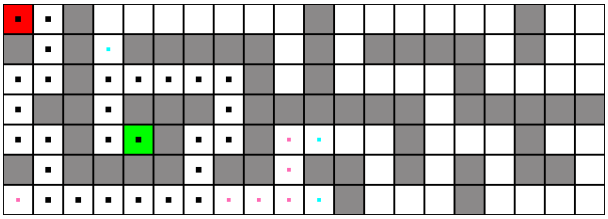

Figure 1.4: Board 1.4

Subproblem A.2: Grids with different cell costs

For this subproblem, we modified our implementation to be able to parse weighted boards as well as correctly handling cost calculation of weighted nodes. Again, the visualizations include the OPEN and CLOSED sets.

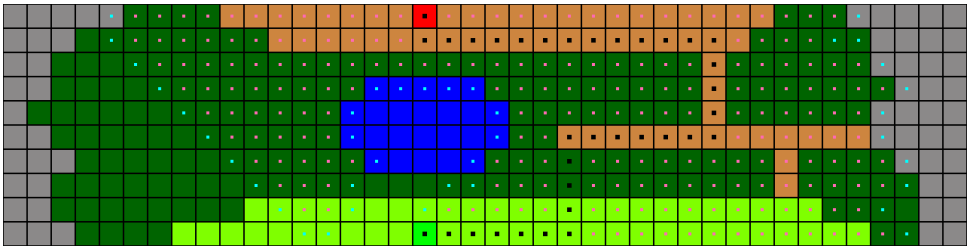

Figure 1.5: Board 2.1

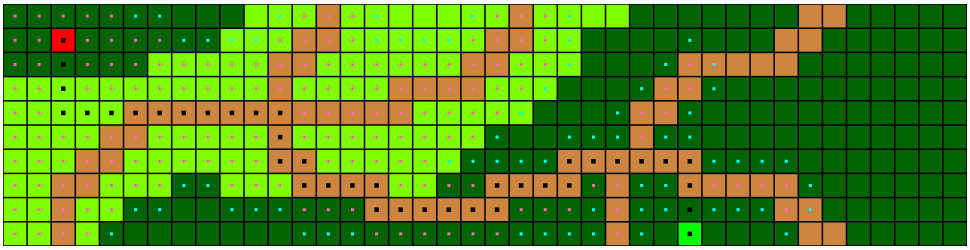

Figure 1.6: Board 2.2

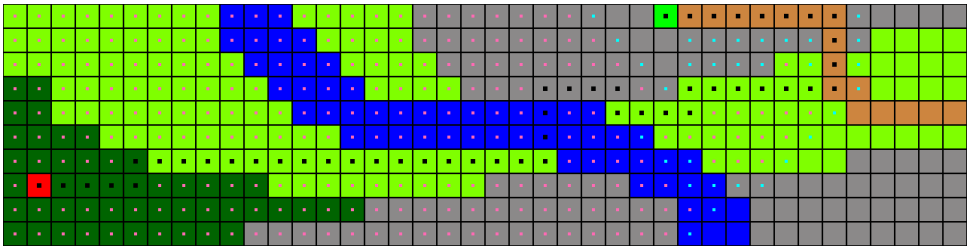

Figure 1.7: Board 2.3

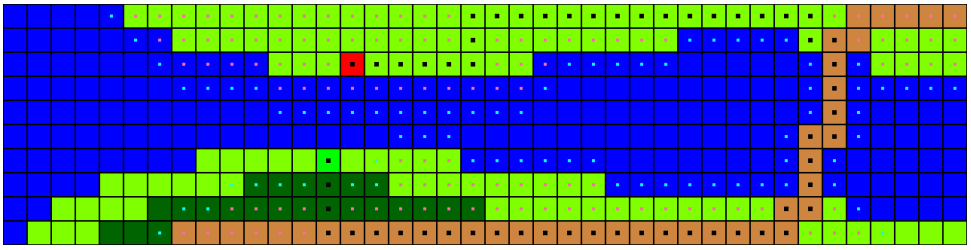

Figure 1.8: Board 2.4

Comparison with BFS and Dijkstra's Algorithm

Board 1.1

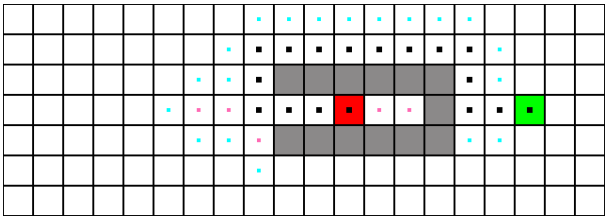

Figure 1.9: Board 1.1 - A*

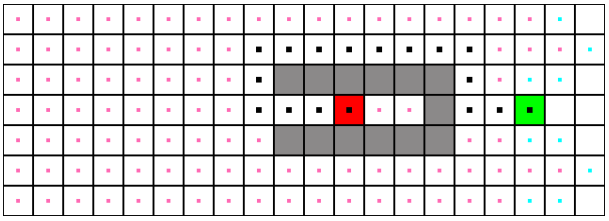

Figure 1.10: Board 1.1 - Dijkstra

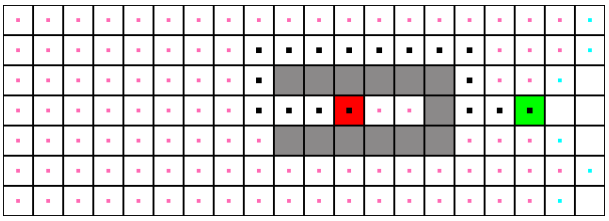

Figure 1.11: Board 1.1 - BFS

All three algorithms find an equally short path, but A* examines significantly fewer nodes doing it.

Board 1.2

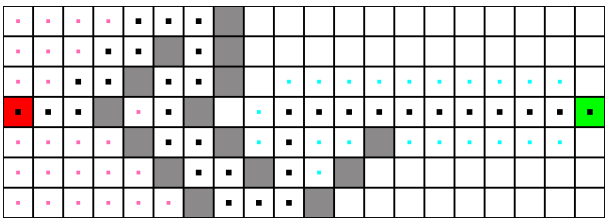

Figure 1.12: Board 1.2 - A*

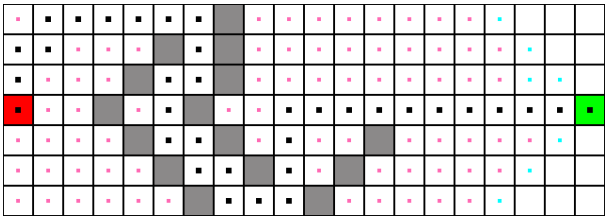

Figure 1.13: Board 1.2 - Dijkstra

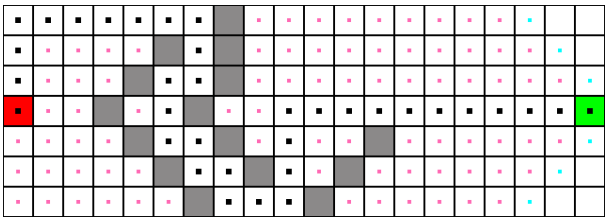

Figure 1.14: Board 1.2 - BFS

All algorithms find the shortest path, but again, A* is much more efficient.

Board 1.3

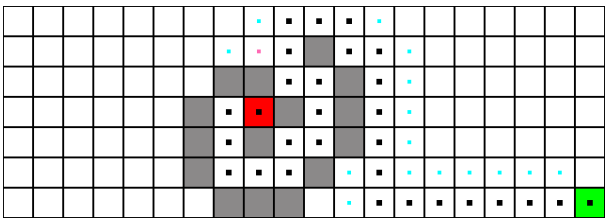

Figure 1.15: Board 1.3 - A*

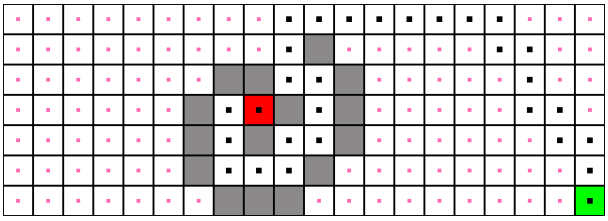

Figure 1.16: Board 1.3 - Dijkstra

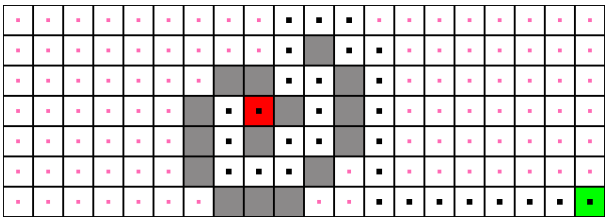

Figure 1.17: Board 1.3 - BFS

Again, A* is much more efficient than the others.

Board 1.4

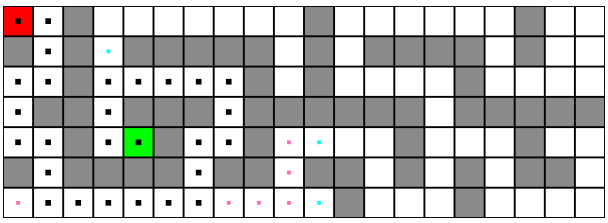

Figure 1.18: Board 1.4 - A*

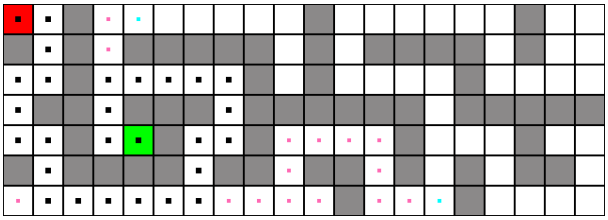

Figure 1.19: Board 1.4 - Dijkstra

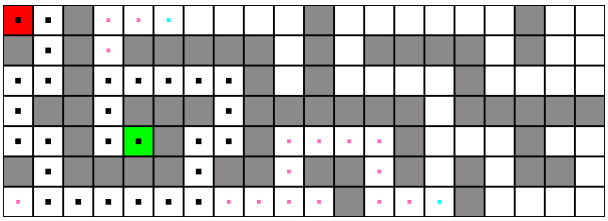

Figure 1.20: Board 1.4 - BFS

For this board, we see that there's not much difference between the three algorithms. A* explores marginally fewer nodes than the other two.